

A large, stylized letter 'A' is formed using the characters 'S' and 'Y'. The 'S' characters are arranged in a grid-like pattern to create the left and right vertical strokes and the horizontal crossbar. The 'Y' characters are used to form the central vertical stroke and the diagonal strokes that meet at the top and bottom. The overall shape is a bold, blocky 'A' that fills most of the page.

[illegible][illegible]


```

(1)      36      HISTORY                                ; DETAILED
(1)      45      DECLARATIONS
(2)      88      TABLES AND CONTROL STRINGS
(3)     151      MMG$PTEDUMP - DUMP PTE AND PFN DATA BASE
(5)     238      SDAS$PTEDUMP - DUMP PTE AND PFN DATA BASE
(6)     299      FORMAT THE PAGE TABLE ENTRY

```

```
0000 1 .TITLE PTEDUMP - DUMP PTE AND PFN DATA BASE
0000 2 .IDENT 'V04-000'
0000 3 :
0000 4 :*****
0000 5 :
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :* ALL RIGHTS RESERVED.
0000 9 :
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :* TRANSFERRED.
0000 16 :
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :* CORPORATION.
0000 20 :
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :
0000 24 :
0000 25 :*****
0000 26 :
0000 27 :++
0000 28 : FACILITY: USER MODE MEMORY MANAGEMENT DIAGNOSTIC ROUTINE
0000 29 :
0000 30 : ABSTRACT:
0000 31 :
0000 32 : ENVIRONMENT: REQUIRES READABILITY OF KERNEL DATA BASES
0000 33 :
0000 34 :--
0000 35 :
0000 36 : .SBTTL HISTORY ; DETAILED
0000 37 :
0000 38 : AUTHOR: PETER H. LIPMAN , CREATION DATE: 19-OCT-76
0000 39 :
0000 40 : MODIFIED BY:
0000 41 : , : VERSION
0000 42 : 01 -
0000 43 :
```



```
0000 45 .SBTTL DECLARATIONS
0000 46
0000 47 :
0000 48 : INCLUDE FILES:
0000 49 :
0000 50 $PCBDEF ;PROCESS CONTROL BLOCK DEFINITIONS
0000 51 $PHDDEF ;PROCESS HEADER DEFINITIONS
0000 52 $PTEDEF ;PAGE TABLE ENTRY DEFINITIONS
0000 53 $SSDEF ;SYSTEM STATUS DEFINITIONS
0000 54 $VADEF ;VIRTUAL ADDRESS DEFINITIONS
0000 55 $WSLDEF ;WORKING SET LIST DEFINITIONS
0000 56 :
0000 57 : EXTERNAL SYMBOLS:
0000 58 :
0000 59 :
0000 60 :
0000 61 : MACROS:
0000 62 :
0000 63 :
0000 64 : EQUATED SYMBOLS:
0000 65 :
0000 66 : OFFSETS FROM AP
0000 67 :
00000000 0000 68 PARAMCNT = 0 ;PARAMETER COUNT
00000004 0000 69 VA = 4 ;VIRTUAL ADDRESS
00000008 0000 70 ADRBYTECNT = 8 ;ADR OF RETURN BYTE COUNT
0000000C 0000 71 OUTSTRING = 12 ;ADR OF OUTPUT STRING DESCRIPTOR
00000010 0000 72 PROCESSINDEX = 16 ;PROCESS INDEX
00000010 0000 73 PTEADR = 16 ;PAGE TABLE ENTRY ADDRESS
00000014 0000 74 PHDADR = 20 ;PROCESS HEADER ADDRESS
00000018 0000 75 PFNBASE = 24 ;BASE OF PFN DATA
0000001C 0000 76 MAXPAGCNT = 28 ;SGNSC MAXPAGCNT
0000 77 ;NO. OF PAGES COVERED BY PFN DATA BASE
0000 78 :
0000 79 : OFFSETS FROM FP
0000 80 :
FFFFFFFFC 0000 81 FAOARG = -4 ;FAO ARGUMENT LIST POINTER
FFFFFFFF8 0000 82 PTEINDEX = -8 ;SAVED INDEX TO PAGE TABLE ENTRY
0000 83 :
0000 84 : OWN STORAGE:
0000 85 :
00000000 86 .PSECT YEXEPAGED ;PAGED PSECT
```



```
0000 88 .SBTTL TABLES AND CONTROL STRINGS
0000 89 :
0000 90 : PROTECTION CODE TABLE
0000 91 :
0000 92 PROTECTIONTBL:
20 20 41 4E 0000 93 .ASCII /NA /
2A 2A 2A 2A 0004 94 .ASCII /****/
20 20 57 4B 0008 95 .ASCII /KW /
20 20 52 4B 000C 96 .ASCII /KR /
20 20 57 55 0010 97 .ASCII /UW /
20 20 57 45 0014 98 .ASCII /EW /
57 4B 52 45 0018 99 .ASCII /ERKW/
20 20 52 45 001C 100 .ASCII /ER /
20 20 57 53 0020 101 .ASCII /SW /
57 45 52 53 0024 102 .ASCII /SREW/
57 4B 52 53 0028 103 .ASCII /SRKW/
20 20 52 53 002C 104 .ASCII /SR /
57 53 52 55 0030 105 .ASCII /URSW/
57 45 52 55 0034 106 .ASCII /UREW/
57 4B 52 55 0038 107 .ASCII /URKW/
20 20 52 55 003C 108 .ASCII /UR /
0040 109 :
0040 110 : OWNER FIELD
0040 111 :
0040 112 OWNERTBL:
55 53 45 4B 0040 113 .ASCII /KESU/
0044 114 :
0044 115 : PAGE TABLE ENTRY TYPE
0044 116 :
0044 117 PTETYPETBL:
53 4E 41 52 54 0044 118 .ASCII /TRANS/
20 58 54 50 47 0049 119 .ASCII /GPTX /
4C 49 46 47 50 004E 120 .ASCII /PGFIL/
20 20 58 54 53 0053 121 .ASCII /STX /
20 4F 52 5A 44 0058 122 .ASCII /DZRO /
44 49 4C 41 56 005D 123 .ASCII /VALID/
47 41 50 4F 49 0062 124 .ASCII /IOPAG/
0067 125 :
0067 126 : MODIFY FLAG TABLE
0067 127 :
0067 128 MODIFYTBL:
4D 20 0067 129 .ASCII / M/
0069 130 :
0069 131 : LOCKED IN WORKING SET LIST TABLE
0069 132 :
0069 133 WSLOCKTBL:
4C 20 0069 134 .ASCII / L/
006B 135 :
006B 136 : FAO CONTROL STRINGS
006B 137 :
006B 138 PTECTL:
44 41 21 28 20 4C 58 21 20 4C 58 21 006B 139 .ASCII /!XL !XL (!AD !AD !AD !AD !AD)/
44 41 21 20 44 41 21 20 44 41 21 20 0077
29 44 41 21 20 0083
0000001D 0088 140 PTECTLSIZ=.-PTECTL
0088 141
55 35 21 20 42 58 21 20 42 58 21 20 0088 142 .ASCII / !XB !XB !SUW !XL !XL !XW !XW/
```


PTEDUMP
V04-000

- DUMP PTE AND PFN DATA BASE
TABLES AND CONTROL STRINGS

E 13

16-SEP-1984 02:38:09 VAX/VMS Macro V04-00
5-SEP-1984 03:46:50 [SYS.SRC]PTEDUMP.MAR;1

Page 4
(2)

58 21 20 4C 58 21 20 4C 58 21 20 57 0094
57 58 21 20 57 00A0
0000003A 00A5
4C' 00A5
00A6
20 20 20 20 20 20 20 41 56 20 20 20 00A6
45 4C 42 41 54 20 45 47 41 50 20 20 00B2
20 20 20 20 20 20 59 52 54 4E 45 20 00BE
20 00CA
20 20 46 45 52 20 20 20 53 54 53 20 00CB
56 53 20 20 20 20 20 4B 41 42 20 20 00D7
46 20 4B 4E 4C 42 20 20 45 54 50 41 00E3
4B 4E 4C 00EF
00F2

143 PFNCTLSIZ=-PTECTL

144 MMG\$PTEDUMPHDR::

145 .BYTE 20\$-10\$

146 10\$:

147 .ASCII \$ VA

PAGE TABLE ENTRY \$

148 .ASCII \$ STS REF BAK SVAPTE BLNK FLNK\$

149 20\$:

```
00F2 151 .SBTTL MMG$PTEDUMP - DUMP PTE AND PFN DATA BASE
00F2 152
00F2 153 :++
00F2 154 : FUNCTIONAL DESCRIPTION:
00F2 155 :
00F2 156 : THIS ROUTINE FORMATS A SINGLE LINE (WITH NO LEADING OR TRAILING
00F2 157 : CR,LF) INTO THE BUFFER SPECIFIED BY THE ADDRESS OF A STRING DESCRIPTOR
00F2 158 : THE TEXT DESCRIBES THE PAGE TABLE ENTRY AND ITS PFN DATA BASE IF ANY
00F2 159 : FOR THE SPECIFIED VA IN THE SPECIFIED PROCESS.
00F2 160 :
00F2 161 :
00F2 162 : CALLING SEQUENCE:
00F2 163 :
00F2 164 : CALLG ARGPTR,MMG$PTEDUMP
00F2 165 :
00F2 166 : INPUT PARAMETERS:
00F2 167 :
00F2 168 : VA(AP) = VIRTUAL ADDRESS
00F2 169 : ADRBYTECNT(AP) = ADDRESS TO RETURN SIZE IN BYTES OF OUTPUT STRING
00F2 170 : OUTSTRING(AP) = DESCRIPTOR OF OUTPUT STRING
00F2 171 : PROCESSINDEX(AP)= PROCESS INDEX IN LOW 16 BITS
00F2 172 :
00F2 173 : IMPLICIT INPUTS:
00F2 174 :
00F2 175 : NONE
00F2 176 :
00F2 177 : OUTPUT PARAMETERS:
00F2 178 :
00F2 179 : RO = SYSTEM STATUS
00F2 180 : @ADRBYTECNT(AP) = SIZE OF STRING FORMATTED OR 0 IF NOTHING DONE
00F2 181 :
00F2 182 : IMPLICIT OUTPUTS:
00F2 183 :
00F2 184 : NONE
00F2 185 :
00F2 186 : COMPLETION CODES:
00F2 187 :
00F2 188 : SSS_NORMAL ;SUCCESSFUL COMPLETION
00F2 189 : SSS_LENvio ;LENGTH VIOLATION
00F2 190 : ;VA IS OFF THE END OF THE PAGE TABLE
00F2 191 :
00F2 192 : SIDE EFFECTS:
00F2 193 :
00F2 194 : NONE
00F2 195 :
00F2 196 :--
```



```
MMGSPTEDUMP::
198
199 .WORD ^M<R2,R3,R4,R5,R6,R7> ;VIRTUAL ADDRESS
200 MOVL VA(AP),R2 ;GET THE PROCESS INDEX
201 MOVZWL PROCESSINDEX(AP),R0 ;OFFSET FROM PHD TO POPT
202 MOVZWL #SGNSC_PHPDAGCNT@7,R1 ;ASSUMING P0 SPACE VIRTUAL ADDRESS
203 ;LEGAL?
204 CMPL R0,#MMGSC_SYSPHX ;BRANCH IF NO, SAY NO PAGES
205 BGTR 40$ ;BRANCH IF NOT SYSTEM PCB
206 BLSS 10$ ;OFFSET FROM SYSPHD TO SYSPT
207 MOVZWL #MMGSC_SYSPHDLEN@-2,R1 ;JAM SYSTEM BIT IN VIRTUAL ADR
208 BBSS #31,R2,10$
209 10$: MOVL SCH$AL_PCB[R0],R4 ;GET PCB ADDRESS
210
211 ; R4 = PCB ADDRESS, R2 = VIRTUAL ADDRESS, R1 = P0 OR SYSTEM SPACE OFFSET
212 ; ALL PROCESS HEADER REFERENCES MUST INDIRECT THROUGH PCB SINCE THIS
213 ; CODE IS SWAPPABLE.
214
215
216 EXTV #VASV_VPN,#VASS_VPN+1,R2,R3 ;SIGN EXTENDED VIRT PAGE NUMBER
217 ;INCLUDING P1 SPACE BIT
218 BLSS 20$ ;BRANCH IF P1 SPACE
219 MOVZBL #PHD$L_FREPOVA@-2,R0 ;LONG WORD INDEX TO FREPOVA
220 CMPL R2,@PCBSL_PHD(R4)[R0] ;ADR BEYOND END OF P0 SPACE
221 BLSSU 30$ ;BRANCH IF NOT LENGTH VIOLATION
222 BRB 40$
223
224 ; P1 SPACE VIRTUAL ADDRESS
225
226 20$: MOVZBL #PHD$L_FREP1VA@-2,R0 ;LONG WORD INDEX TO FREP1VA
227 CMPL R2,@PCBSL_PHD(R4)[R0] ;OFF THE END OF P1 SPACE?
228 BLEQU 40$ ;BRANCH IF YES, LENGTH VIOLATION
229 MOVZWL #<SGNSC_PHPDAGCNT+SGNSC_PTPAGCNT>@7,R1 ;OFFSET TO P1PT END
230 30$: ADDL R1,R3 ;INDEX TO PTE
231 MOVAL PCBSL_PHD(R4),R5 ;POINTER TO PROCESS HEADER ADDRESS
232 MOVAL PFNSA_BASE,R6 ;BASE OF PFN DATA
233 MOVL #SGNSC_MAXPAGCNT,R7 ;NO. OF PAGES COVERED BY PFN DATA BASE
234 BRB ;GO FORMAT THE PAGE TABLE ENTRY
235 40$: MOVZWL #SS$_LENVI0,R0 ;LENGTH VIOLATION
236 BRB NULLSTRING
```

```
015D 238 .SBTTL SDA$PTEDUMP - DUMP PTE AND PFN DATA BASE
015D 239
015D 240 :++
015D 241 : FUNCTIONAL DESCRIPTION:
015D 242 :
015D 243 : THIS ROUTINE FORMATS A SINGLE LINE (WITH NO LEADING OR TRAILING
015D 244 : CR,LF) INTO THE BUFFER SPECIFIED BY THE ADDRESS OF A STRING DESCRIPTOR
015D 245 : THE TEXT DESCRIBES THE PAGE TABLE ENTRY AND ITS PFN DATA BASE IF ANY
015D 246 : FOR THE SPECIFIED VA.
015D 247 : IT PROVIDES THE SYSTEM DUMP ANALYZER WITH A STANDARD FORMATTING
015D 248 : ROUTINE WHICH IS INDEPENDENT OF SYSTEM GENERATION DIFFERENCES.
015D 249 :
015D 250 : CALLING SEQUENCE:
015D 251 :
015D 252 : CALLG ARGPTR,SDA$PTEDUMP
015D 253 :
015D 254 : INPUT PARAMETERS:
015D 255 :
015D 256 : VA(AP) = VIRTUAL ADDRESS
015D 257 : ADRBYTECNT(AP) = ADDRESS TO RETURN SIZE IN BYTES OF OUTPUT STRING
015D 258 : OUTSTRING(AP) = DESCRIPTOR OF OUTPUT STRING
015D 259 : PTEADR(AP) = PAGE TABLE ENTRY ADDRESS
015D 260 : PHDADR(AP) = PROCESS HEADER ADDRESS
015D 261 : PFNBASE(AP) = PFN$A_BASE - BASE ADDRESS OF PFN DATA BASE
015D 262 : MAXPAGCNT(AP) = SGN$C_MAXPAGCNT - PAGES COVERED BY PFN DATA BASE
015D 263 :
015D 264 : IMPLICIT INPUTS:
015D 265 :
015D 266 : NONE
015D 267 :
015D 268 : OUTPUT PARAMETERS:
015D 269 :
015D 270 : RO = SYSTEM STATUS
015D 271 : @ADRBYTECNT(AP) = SIZE OF STRING FORMATTED OR 0 IF NOTHING DONE
015D 272 :
015D 273 : IMPLICIT OUTPUTS:
015D 274 :
015D 275 : NONE
015D 276 :
015D 277 : COMPLETION CODES:
015D 278 :
015D 279 : SS$_NORMAL ;SUCCESSFUL COMPLETION
015D 280 :
015D 281 : SIDE EFFECTS:
015D 282 :
015D 283 : NONE
015D 284 :
015D 285 : --
015D 286 :
015D 287 SDA$PTEDUMP::
015D 288 .WORD ^M<R2,R3,R4,R5,R6,R7>
015D 289 MOVL VA(AP),R2 ;VIRTUAL ADDRESS
015D 290 SUBL3 PHDADR(AP),PTEADR(AP),R3 ;FORM INDEX TO PAGE TABLE ENTRY
015D 291 ASHL #-2,R3,R3 ;LONG WORD INDEX
015D 292 MOVAL PHDADR(AP),R5 ;POINTER TO PROCESS HEADER ADDRESS
015D 293 MOVL PFNBASE(AP),R6 ;BASE ADR OF PFN DATA BASE
015D 294 MOVL MAXPAGCNT(AP),R7 ;PAGE COUNT COVERED BY PFN DATA BASE
```

```
53 10 52 04 AC 00FC
53 53 AC 14 AC C3
55 53 FE 8F 78
56 14 AC DE
57 18 AC DO
57 1C AC DO
```


PTEDUMP
V04-000

I 13
- DUMP PTE AND PFN DATA BASE 16-SEP-1984 02:38:09 VAX/VMS Macro V04-00
SDASPTEDUMP - DUMP PTE AND PFN DATA BASE 5-SEP-1984 03:46:50 [SYS.SRC]PTEDUMP.MAR;1

Page 8
(5)

017A 295 :
017A 296 : FALL THROUGH TO FORMAT PTE
017A 297 :

```
017A 299 .SBTTL FORMAT THE PAGE TABLE ENTRY
017A 300 :
017A 301 : R2 = VIRTUAL ADDRESS
017A 302 : R3 = LONG WORD INDEX TO PAGE TABLE ENTRY
017A 303 : R5 = POINTER TO PROCESS HEADER ADDRESS
017A 304 : R6 = BASE OF PFN DATA ARRAY
017A 305 : R7 = NO. OF PAGES COVERED BY THE PFN DATA BASE
017A 306 :
017A 307 .ENABL LSB
017A 308 FORMATPTE:
017A 309 SUBL3 #21*4,FP,SP ;RESERVE FAO PARAMETER BUFFER
0182 310 MOVL SP,FAOARG(FP) ;AND SAVE ITS ADDRESS
0186 311 MOVL R3,PTEINDEX(FP) ;SAVE INDEX TO PTE
018A 312 MOVL SP,R4 ;R4 USED TO STORE FAO PARAMS
018D 313 :
018D 314 : SET UP TO CONVERT VA, PTE, AND FORMATTED PTE
018D 315 :
018D 316 MOVL R2,(R4)+ ;VIRTUAL ADDRESS
0190 317 MOVL @R5[R3],R3 ;R3 = PAGE TABLE ENTRY
0195 318 BNEQ 30$ ;BRANCH IF NOT DELETED
0197 319 MOVZWL #SS$_NORMAL,R0 ;NO OUTPUT FOR 0 PTE'S
019A 320 NULLSTRING:
019A 321 CLRL @ADRBYTECNT(AP) ;RETURN STRING IS 0 BYTES
019D 322 RET
019E 323 30$:
019E 324 EXTZV #PTESV_PFN,#PTESS_PFN,R3,R0 ;R0 = PFN IF PRESENT
01A3 325 MOVL #5,R1 ;ASSUME VALID PTE TYPE
01A6 326 MOVL R3,(R4)+ ;CONVERT PTE IN HEX
01A9 327 BGEQ 45$ ;BRANCH IF NOT VALID
01AB 328 CMPL R0,R7 ;LEGAL PFN?
01AE 329 BLSS 60$ ;YES, VALID PAGE
01B0 330 INCL R1 ;ASSUME IT'S AN I/O PAGE
01B2 331 BRB 50$ ;AND SAY NO PFN
01B4 332 45$:
01B4 333 EXTZV #PTESV_TYPO,#1,R3,R1 ;FORM THE PTE TYPE CODE
01B9 334 BBC #PTESV_TYP1,R3,40$ ;IF TYPE BIT 1 IS 0 BRANCH
01BD 335 BISL #2,R1 ;SET HIGH ORDER TYPE BIT
01C0 336 40$:
01C0 337 TSTL R1 ;TYPE 0 IS TRANSITION OR DZRO
01C2 338 BNEQ 50$ ;BRANCH IF NOT TYPE 0
01C4 339 TSTL R0 ;PFN = 0 IF DZRO
01C6 340 BNEQ 60$ ;BRANCH IF TRANSITION
01C8 341 MOVL #4,R1 ;TYPE CODE FOR DZRO
01CB 342 50$:
01CB 343 MNEGL #1,R0 ;NO PFN IN THIS PTE
01CE 344 :
01CE 345 : R1 = PTE TYPE CODE, SEE TABLE, R0 = PFN OR -1 IF NONE
01CE 346 :
01CE 347 60$:
01CE 348 MULL #5,R1 ;PTE TYPE CODE IS 5 CHARS WIDE
01D1 349 MOVL #5,(R4)+ ;SIZE OF PTE TYPE STRING
01D4 350 MOVAB PTETYPETBL[R1],(R4)+ ;ADR OF PTE TYPE STRING
01DA 351 CLRQ R1 ;ASSUME NO MODIFY OR LOCK BITS
01DC 352 BBC #PTESV_VALID,R3,70$ ;BRANCH IF PTE NOT VALID
01E0 353 EXTZV #PTESV_MODIFY,#1,R3,R1 ;GET MODIFY BIT FROM PTE
01E5 354 BBS #31,R0,70$ ;BRANCH IF NO PFN
01E9 355 BBS #31,@FAOARG(FP),70$ ;BRANCH IF SYSTEM ADDRESS
```

5E 5D 00000054 8F C3 017A 299
FC AD 5E D0 017A 300
F8 AD 53 D0 017A 301
54 5E D0 017A 302
54 5E D0 017A 303
54 5E D0 017A 304
54 5E D0 017A 305
54 5E D0 017A 306
54 5E D0 017A 307
54 5E D0 017A 308
54 5E D0 017A 309
54 5E D0 0182 310
54 5E D0 0186 311
54 5E D0 018A 312
54 5E D0 018D 313
54 5E D0 018D 314
54 5E D0 018D 315
54 5E D0 018D 316
54 5E D0 0190 317
54 5E D0 0195 318
54 5E D0 0197 319
54 5E D0 019A 320
54 5E D0 019A 321
54 5E D0 019D 322
54 5E D0 019E 323
54 5E D0 019E 324
54 5E D0 01A3 325
54 5E D0 01A6 326
54 5E D0 01A9 327
54 5E D0 01AB 328
54 5E D0 01AE 329
54 5E D0 01B0 330
54 5E D0 01B2 331
54 5E D0 01B4 332
54 5E D0 01B4 333
54 5E D0 01B9 334
54 5E D0 01BD 335
54 5E D0 01C0 336
54 5E D0 01C0 337
54 5E D0 01C2 338
54 5E D0 01C4 339
54 5E D0 01C6 340
54 5E D0 01C8 341
54 5E D0 01CB 342
54 5E D0 01CB 343
54 5E D0 01CE 344
54 5E D0 01CE 345
54 5E D0 01CE 346
54 5E D0 01CE 347
54 5E D0 01CE 348
54 5E D0 01D1 349
54 5E D0 01D4 350
54 5E D0 01DA 351
54 5E D0 01DC 352
54 5E D0 01E0 353
54 5E D0 01E5 354
54 5E D0 01E9 355


```
52 57 00' C5 01EE 356 MULL3 S^#PFNSC_WSLX,R7,R2 ;BYTE OFFSET TO WSLX ARRAY
      52 56 C0 01F2 357 ADDL R6,R2 ;BYTE ADDRESS OF WSLX ARRAY
      52 6240 3C 01F5 358 MOVZWL (R2)[R0],R2 ;WORKING SET LIST INDEX
      0A 13 01F9 359 BEQL 70$ ;BRANCH IF NOT A WSL INDEX
52 52 00 B542 D0 01FB 360 MOVL @ (R5)[R2],R2 ;WORKING SET LIST ENTRY
52 52 01 05 EF 0200 361 EXTZV #WSLSV_WSLOCK,#1,R2,R2 ;WORKING SET LIST LOCK BIT
      0205 362 70$:
      84 01 D0 0205 363 MOVL #1,(R4)+ ;SIZE OF MODIFY STRING
84 FE5A CF41 9E 0208 364 MOVAB MODIFYTBL[R1],(R4)+ ;ADR OF MODIFY STRING
      84 01 D0 020E 365 MOVL #1,(R4)+ ;SIZE OF WRK SET LOCK STRNG
84 FE53 CF42 9E 0211 366 MOVAB WSLOCKTBL[R2],(R4)+ ;ADR OF WRK SET LOCK STRING
      0217 367
51 53 04 1B EF 0217 368 EXTZV #PTESV_PROT,#PTESS_PROT,R3,R1 ;R1 = PROTECTION CODE
      84 04 D0 021C 369 MOVL #4,(R4)+ ;SIZE OF PROTECTION STRING
84 FDDC CF41 DE 021F 370 MOVAL PROTECTIONTBL[R1],(R4)+ ;ADR OF PROTECTION STRING
      0225 371
51 53 02 17 EF 0225 372 EXTZV #PTESV_OWN,#PTESS_OWN,R3,R1 ;R1 = OWNER OF PTE
      84 01 D0 022A 373 MOVL #1,(R4)+ ;SIZE OF OWNER STRING
84 FE0E CF41 9E 022D 374 MOVAB OWNERTBL[R1],(R4)+ ;ADR OF OWNER STRING
      0233 375
      0233 376 : NOW SET UP TO CONVERT THE PFN DATA BASE IF ANY
      0233 377 :
      FE34 CF DF 0233 378 PUSHAL PTECTL ;FORM CONTROL STRING DESCRIPTOR
      1D DD 0237 379 PUSHL #PTECTLSIZ ;ON THE STACK
      51 5E D0 0239 380 MOVL SP,R1 ;R1 = ADDRESS OF STRING DESCR
50 50 1F E0 023C 381 BBS #31,R0,80$ ;BRANCH IF NO PFN
      61 3A 9A 0240 382 MOVZBL #PFNCTLSIZ,(R1) ;INCLUDE PFN CONTROL STRING
52 57 00' C5 0243 383 MULL3 S^#PFNSC_STATE,R7,R2 ;BYTE OFFSET TO STATE ARRAY
      52 56 C0 0247 384 ADDL R6,R2 ;ADDRESS OF STATE ARRAY
      84 6240 9A 024A 385 MOVZBL (R2)[R0],(R4)+ ;STATE BYTE
52 57 00' C5 024E 386 MULL3 S^#PFNSC_TYPE,R7,R2 ;BYTE OFFSET TO TYPE ARRAY
      52 56 C0 0252 387 ADDL R6,R2 ;ADDRESS OF TYPE ARRAY
      84 6240 9A 0255 388 MOVZBL (R2)[R0],(R4)+ ;TYPE BYTE
52 57 00' C5 0259 389 MULL3 S^#PFNSC_REFCNT,R7,R2 ;BYTE OFFSET TO REFCNT ARRAY
      52 56 C0 025D 390 ADDL R6,R2 ;ADDRESS OF REFCNT ARRAY
      84 6240 3C 0260 391 MOVZWL (R2)[R0],(R4)+ ;REFERENCE COUNT
52 57 00' C5 0264 392 MULL3 S^#PFNSC_BAK,R7,R2 ;BYTE ADDRESS OF BACKING STORE ARRAY
      52 56 C0 0268 393 ADDL R6,R2 ;ADDRESS OF BACKING STORE ARRAY
      84 6240 D0 026B 394 MOVL (R2)[R0],(R4)+ ;BACKING STORE ADDRESS
52 57 00' C5 026F 395 MULL3 S^#PFNSC_PTE,R7,R2 ;BYTE OFFSET TO PTE ARRAY
      52 56 C0 0273 396 ADDL R6,R2 ;ADDRESS OF PTE ARRAY
      84 6240 D0 0276 397 MOVL (R2)[R0],(R4)+ ;PTE POINTER
52 57 00' C5 027A 398 MULL3 S^#PFNSC_BLINK,R7,R2 ;BYTE OFFSET TO BLINK ARRAY
      52 56 C0 027E 399 ADDL R6,R2 ;ADDRESS OF BLINK ARRAY
      84 6240 3C 0281 400 MOVZWL (R2)[R0],(R4)+ ;BACK LINK
52 57 00' C5 0285 401 MULL3 S^#PFNSC_FLINK,R7,R2 ;BYTE OFFSET TO FLINK ARRAY
      52 56 C0 0289 402 ADDL R6,R2 ;ADDRESS OF FLINK ARRAY
      84 6240 3C 028C 403 MOVZWL (R2)[R0],(R4)+ ;FORWARD LINK
      0290 404 80$:
      50 53 D0 0290 405 MOVL R3,R0 ;PAGE TABLE ENTRY CONTENTS
53 F8 AD D0 0293 406 MOVL PTEINDEX(FP),R3 ;INDEX TO PAGE TABLE ENTRY
50 00 B543 D1 0297 407 CMPL @ (R5)[R3],R0 ;STILL THE SAME PTE?
      07 13 029C 408 BEQL 90$ ;BRANCH IF YES
      52 04 AC D0 029E 409 MOVL VA(AP),R2 ;MUST RETRY, GET VA
      FED5 31 02A2 410 BRW FORMATPTE ;START OVER AGAIN
      02A5 411 90$:
      02A5 412 $FAOL_S (R1),@ADRBYTECNT(AP),@OUTSTRING(AP),@FAOARG(FP)
```

PTEDUMP
V04-000

- DUMP PTE AND PFN DATA BASE
FORMAT THE PAGE TABLE ENTRY

L 13

16-SEP-1984 02:38:09 VAX/VMS Macro V04-00
5-SEP-1984 03:46:50 [SYS.SRC]PTEDUMP.MAR;1

Page 11
(6)

04	02B7	413	RET
	02B8	414	
	02B8	415	.DSABL LSB
	02B8	416	
	02B8	417	
	02B8	418	.END

PTEDUMP
Symbol table

- DUMP PTE AND PFN DATA BASE

M 13

16-SEP-1984 02:38:09 VAX/VMS Macro V04-00
5-SEP-1984 03:46:50 [SYS.SRC]PTEDUMP.MAR;1

Page 12
(6)

ADRBYTECNT	= 00000008		
FAOARG	= FFFFFFFC		
FORMATPTE	0000017A	R	02
MAXPAGCNT	= 0000001C		
MMGSC_SYSPHDLEN	*****	X	02
MMGSC_SYSPIX	*****	X	02
MMGSPTEDUMP	000000F2	RG	02
MMGSPTEDUMPHDR	000000A5	RG	02
MODIFYTBL	00000067	R	02
NULLSTRING	0000019A	R	02
OUTSTRING	= 0000000C		
OWNERTBL	00000040	R	02
PARAMCNT	= 00000000		
PCBSL_PHD	= 0000006C		
PFNSA_BASE	*****	X	02
PFNSC_BAK	*****	X	02
PFNSC_BLINK	*****	X	02
PFNSC_FLINK	*****	X	02
PFNSC_PTE	*****	X	02
PFNSC_REFCNT	*****	X	02
PFNSC_STATE	*****	X	02
PFNSC_TYPE	*****	X	02
PFNSC_WSLX	*****	X	02
PFNBASE	= 00000018		
PFNCTLSIZ	= 0000003A		
PHDSL_FREPOVA	= 00000028		
PHDSL_FREPIVA	= 00000030		
PHDADR	= 00000014		
PROCESSINDEX	= 00000010		
PROTECTIONTBL	00000000	R	02
PTES\$OWN	= 00000002		
PTES\$PFN	= 00000015		
PTES\$PROT	= 00000004		
PTESV_MODIFY	= 0000001A		
PTESV_OWN	= 00000017		
PTESV_PFN	= 00000000		
PTESV_PROT	= 0000001B		
PTESV_TYPO	= 00000016		
PTESV_TYPI	= 0000001A		
PTESV_VALID	= 0000001F		
PTEADR	= 00000010		
PTECTL	0000006B	R	02
PTECTLSIZ	= 0000001D		
PTEINDEX	= FFFFFFF8		
PTETYPETBL	00000044	R	02
SCH\$AL_PCB	*****	X	02
SDASPTEDUMP	0000015D	RG	02
SGNSC_MAXPAGCNT	*****	X	02
SGNSC_PHDPAGCNT	*****	X	02
SGNSC_PTPAGCNT	*****	X	02
SS\$_LENVIO	= 0000018C		
SS\$_NORMAL	= 00000001		
SY\$FAOL	*****	GX	02
VA	= 00000004		
VAS\$VPN	= 00000015		
VASV_VPN	= 00000009		
WSL\$V_WSLOCK	= 00000005		

WSLOCKTBL

00000069 R 02

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes															
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE					
\$ABS\$	00000000 (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					
YEXEPAGED	000002B8 (696.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.27
Command processing	105	00:00:00.59	00:00:01.23
Pass 1	271	00:00:07.35	00:00:08.63
Symbol table sort	0	00:00:01.19	00:00:01.22
Pass 2	86	00:00:01.65	00:00:01.82
Symbol table output	9	00:00:00.08	00:00:00.08
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	504	00:00:10.99	00:00:13.28

The working set limit was 1500 pages.

43385 bytes (85 pages) of virtual memory were used to buffer the intermediate code.

There were 50 pages of symbol table space allocated to hold 788 non-local and 14 local symbols.

418 source lines were read in Pass 1, producing 15 object records in Pass 2.

15 pages of virtual memory were used to define 14 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	5
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	11

845 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:PTEDUMP/OBJ=OBJ\$:PTEDUMP MSRC\$:PTEDUMP/UPDATE=(ENH\$:PTEDUMP)+EXECML\$/LIB

0379

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY